Enabling adaptive scientific workflows via trigger detection*

Maher Salloum Janine C. Bennett Ali Pinar
Ankit Bhagatwala Jacqueline H. Chen
Sandia National Laboratories, Livermore, CA
{mnsallo, jcbenne, apinar, abhagat, jhcehn} @sandia.gov

ABSTRACT

Next generation architectures necessitate a shift away from traditional workflows in which the simulation state is saved at prescribed frequencies for post-processing analysis. While the need to shift to in situ workflows has been acknowledged for some time, much of the current research is focused on *static workflows*, where the analysis that would have been done as a post-process is performed concurrently with the simulation at user-prescribed frequencies. Recently, research efforts are striving to enable *adaptive workflows*, in which the frequency, composition, and execution of computational and data manipulation steps dynamically depend on the state of the simulation. Adapting the workflow to the state of simulation in such a data-driven fashion puts extremely strict efficiency requirements on the analysis capabilities that are used to identify the transitions in the workflow. In this paper we build upon earlier work on trigger detection using sublinear techniques to drive adaptive workflows. Here we propose a methodology to detect the time when sudden heat release occurs in simulations of turbulent combustion. Our proposed method provides an alternative metric that can be used along with our former metric to increase the robustness of trigger detection. We show the effectiveness of our metric empirically for predicting heat release for two use cases.

1. INTRODUCTION

Concurrent analysis frameworks have been developed to process raw simulation output as it is computed, decoupling the analysis from I/O. Operations sharing primary resources of the simulation are considered *in situ*, while *in transit* processing involves asynchronous data transfers to secondary resources. Both in situ [15, 6, 8] and in transit [14, 1, 2] workflows perform analyses as the simulation is run and produce results, which are typically much smaller than the raw data, mitigating the effects of limited disk bandwidth and capacity. Concurrent analyses are often performed at prespecified frequencies, which is viable for analyses that are not too expensive – in terms of runtime (with respect to a simulation time step), memory footprint, and output size. However, for many other analyses that are resource intense, prescribed frequencies will not suffice because the scientific phenomenon being simulated typically does not behave linearly (e.g., combustion, climate, astrophysics). When the prescribed I/O or analysis frequency is high enough to capture the events of interest, the costs incurred are too great. On the other hand, a cost-effective, lower analysis-frequency may miss the scientific events that simulation is intended to capture.

An alternative approach is to perform expensive analyses (denoted A_e) and I/O in an adaptive fashion, driven by the data itself. A domain-agnostic approach is presented in [12, 11], based on entropy of information in the data. Our previous work in [3] provides a framework for making data-driven control-flow decisions that can leverage the scientists' intuitions, even when the algorithms to capture those intuitions would otherwise be too expensive to compute. Our methodology involves a user-defined *indicator* function that is computed and measured in situ at a relatively regular and high-frequency (i.e., greater than the frequency with which the I/O or A_e would be prescribed). Along with the indicator, the application scientist defines an associated *trigger*, a function that returns a boolean value indicating that the indicator has met some property, for example a threshold value. While our methodology is intuitive and conceptually quite simple, the challenges lie in defining indicators and triggers that capture the appropriate scientific information, while remaining cost efficient in terms of runtime, memory footprint, and I/O requirements, so that they can be deployed at the high frequency that is required.

In [3] we show that chemical explosive mode analysis, denoted CEMA, can be used to devise a noise-tolerant indicator for heat release (the quantity of interest that the scientists would like to capture), thus making it a good candidate to drive adaptive workflows. However, exhaustive computation of CEMA values dominates the simulation time. To overcome this bottleneck, we proposed a quantile sampling approach with provable error/confidence bounds. These bounds depend only on the number of samples and are independent of the problem size. We also de-

*This work was funded by the Laboratory Directed Research and Development (LDRD) program of Sandia National Laboratories. Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

signed an indicator, referred to as the *P-indicator*, based on the quantile sampling approach. Our experiments on homogeneous charge compression ignition (HCCI) and reactivity controlled compression ignition (RCCI) simulations showed that the proposed method can detect rapid increases in heat release *and* is computational efficient.

In this paper, we propose an alternative indicator, referred to as the *C-indicator*, and associated trigger function. The proposed technique is based on the coefficient of variation and similar in essence to our earlier technique that used quantiles, as it tries to detect shrinking in the range of upper percentiles in the distribution of CEMA values. The coefficient of variation among the top quantiles decreases as the range of the values shrink and can serve as a detector for the subsequent heat release that the scientists wish to capture. The new technique based on coefficient of variation is not proposed as an alternative to replace our former metric. Rather, we believe that a set of trigger detection mechanisms, when used collectively, can provide more robust detections, as collectively they will be less prone to false positives. Our experiments on HCCI and RCCI simulations show that the proposed method can efficiently detect rapid increases in heat release.

2. COMBUSTION AS A USE CASE FOR TRIGGER DETECTION

We demonstrate our approach applied to a combustion use case, using S3D [7], a direct numerical simulation of combustion in turbulence. The combustion simulations in our use case pertain to a class of internal combustion (IC) engine concept. We are interested in two specific techniques called homogeneous-charge compression ignition (HCCI) [4] and reactivity-controlled compression ignition (RCCI) [9, 5]. In both cases, heat release starts in the form of small kernels at arbitrary locations. Eventually, multiple kernels ignite as the overall heat release reaches a global maximum and subsequently declines. Since these simulations are computation and storage-intense, we want to run the simulation at a coarser grid resolution and save data less frequently during the early build-up phase. When the heat release events occur, we want to run the simulation at the finest grid granularity possible, and store the data as frequently as possible. Therefore, it is imperative to be able to predict the start of the heat release event using an indicator and trigger that serve to inform the application to adjust its grid resolution and I/O frequency accordingly, see Figure 1.

3. DESIGNING A NOISE-RESISTANT INDICATOR AND TRIGGER

In this section we describe the design of an indicator and trigger for heat release for our combustion use case. To provide context, we begin by discussing the intuitions that informed our design.

3.1 Chemical Explosive Mode Analysis

Chemical explosive mode analysis (CEMA) is known to be a reliable technique to predict incipient heat release. Here we provide brief description and refer to [10, 13] for details. The conservation equations for reacting species can be written as $\frac{D\mathbf{y}}{Dt} = \omega(\mathbf{y}) + \mathbf{s}(\mathbf{y})$. The vector $\mathbf{y}$ represents temperature and reacting species mass fractions, ω is the reaction source term and $\mathbf{s}$ is the mixing term. Then, the

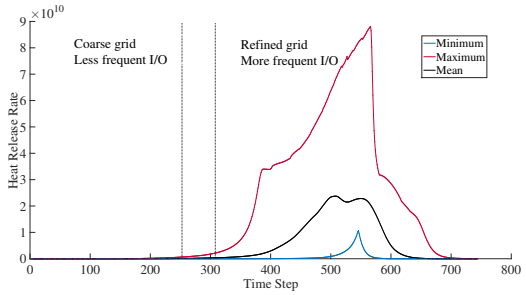

Figure 1: The minimum (blue), maximum (red) and mean (black) heat release values for each time step in the simulation. Early in the simulation, we want to run at a coarser grid and save data less frequently. When heat release occurs, we want to the finest grid and save the data as frequently as possible. The vertical lines define a range of time steps within which we would like to make this transition, as identified by a domain expert.

Jacobian is $\mathbf{J}_\omega + \mathbf{J}_\mathbf{s}$ where $\mathbf{J}_\omega = \frac{\partial \omega(\mathbf{y})}{\partial \mathbf{y}}$ and $\mathbf{J}_\mathbf{s} = \frac{\partial \mathbf{s}(\mathbf{y})}{\partial \mathbf{y}}$. The eigen-decomposition of the chemical Jacobian, $\mathbf{J}_\omega$ can be used to infer chemical properties of the mixture. Let λ_e be the eigenvalue with the largest real part. λ_e is defined as a chemical explosive mode (CEM) if $\mathrm{Re}(\lambda_e) > 0$, which indicates that point will undergo ignition. If it has undergone ignition, we have $Re(\lambda_e) < 0$. The presence of a CEM indicates the propensity of a mixture to ignite.

Our CEMA-based indicator is based on global trends of CEMA over time. Consider Figure 2 which provides a summary of the trends of CEMA values across all time steps in a simulation. At timestep t, let $\mathcal{C}(t)$ be the sorted (in non-decreasing order) array of CEMA values on the underlying mesh. For $\alpha \in (0, 1]$, the α-percentile is the entry $\mathcal{C}(t)_{\lceil \alpha N \rceil}$. More specifically, it is the value in $\mathcal{C}(t)$ that is greater than at least $\lceil \alpha N \rceil$ values in $\mathcal{C}(t)$. We denote this value by $p_\alpha(t)$.

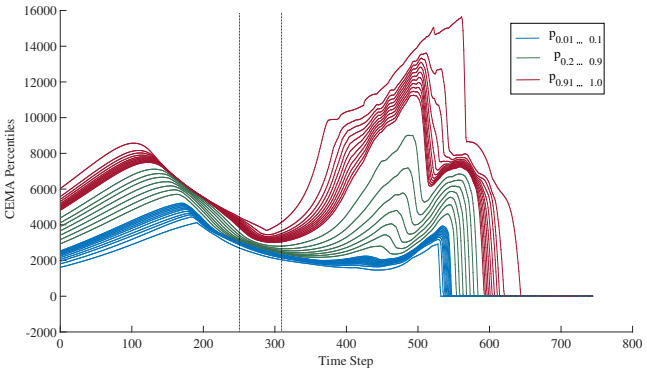

Figure 2: Percentile plot of CEMA values. The blue curves correspond to $p_{0.01,0.02,\ldots,0.1}$, the green curves to $p_{0.2,0.3,\ldots,0.9}$ and the red curves to $p_{0.91,0.92,\ldots,1}$.

We notice that as the simulation progresses, the distance between the higher percentiles (red curves) decreases then suddenly increases. This is illustrated in the plot by the spread in the red curves that occurs between the dashed lines defined by the domain expert that indicate the desired transition phase. Our goal is to capture this empirical observation with a metric, and use it to predict heat release.

Our empirical observation is consistent with the underlying physics, and we refer to [3] for a detailed discussion on the underlying physics.

3.2 Designing a Noise-Resistant CEMA-Based Indicator

We introduce an indicator function, *C-indicator* that quantifies the distribution of the top quantiles of CEMA values over time. The C-indicator is equal to the coefficient of variation (COV) of the top percentiles of CEMA values, i.e. the ratio of their standard deviation to their mean.

Formally, quantiles are defined as values taken at regular intervals from the inverse of the cumulative distribution function of a random variable. For a given data set, quantiles are used to divide the data into equal sized sets after sorting, and the quantiles are the values on the boundary between consecutive subsets. A special case is dividing into 100 equal groups, when we can refer to quantiles as percentiles. This paper focuses on percentiles with numbers in the $[0, 1]$ range (although all techniques presented here can be generalized for any quantiles). For example, the 0.5 percentile will refer to the median of the data set.

We define our indicator using 2 parameters: α and β. We want to detect whether the range covered by top quantiles shrinks, and α represents the lower end of the top percentiles, whereas β represents the top percentile considered. Therefore, the range of top percentiles we measure is $[p_\alpha(t), p_\beta(t)]$. In our indicator, we choose $\alpha < \beta$ (typically in the range $[0.90, 0.99]$). We measure the spread at time t by the COV of the $p_s(t)$ values for $\alpha \leq s \leq \beta$ which results in the following C-indicator:

$$C_{\alpha,\beta}(t) = \sqrt{\frac{\mu}{N-1}\sum_{s=\alpha}^{\beta}(p_s/\mu - 1)^2} \tag{1}$$

where

$$\mu = \frac{1}{N}\sum_{s=\alpha}^{\beta} p_s \tag{2}$$

and N is the number of percentile values considered between α and β. For instance, if $\alpha = 0.90$ and $\beta = 0.99$, we would obtain $N = 10$.

Figure 3 illustrates percentile plots for heat release (top row) and CEMA (middle row). In the percentile plots, the lowest blue curve and the highest red curve correspond to the 1 and 100 percentiles ($p_{0.01}$ and p_1), respectively. The blue curves correspond to $p_{0.01,0.02,\ldots,0.1}$, the green curves to $p_{0.2,0.3,\ldots,0.9}$ and the red curves to $p_{0.91,0.92,\ldots,1}$. The C-indicator evaluated for $\alpha = 0.92$ and $\beta = 0.99$ is shown in the bottom row of Figure 3. Results are generated for four test cases described in Table 1. The vertical dotted lines were identified by a domain expert who, via examination of heat release and CEMA percentile plots, visually located the range of time steps in the simulation where the mesh resolution and I/O frequency should be increased. We refer to this range of time steps as the "true" trigger range we wish to identify with the C-indicator and trigger functions. Note, for the RCCI cases, there are two ignition ranges. To simplify the following exposition, we focus on the second rise in the heat release rate profiles, as this is the ignition stage of interest to the scientists. However, we note that our approach is robust in identifying the first ignition stage as well.

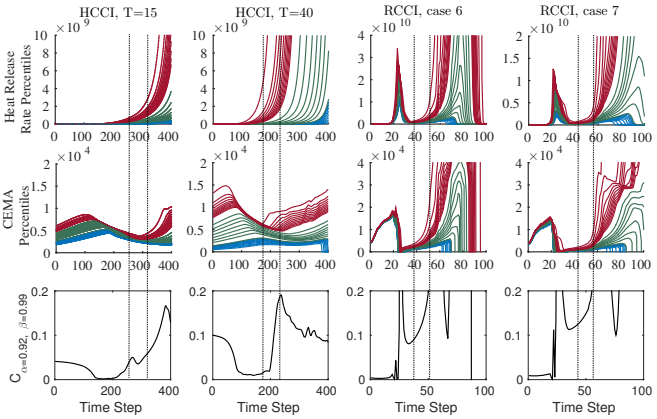

Figure 3: Plots showing the (top row) percentiles of the heat release rate, (middle row) percentiles of CEMA, and (bottom row) the C-indicator, as indicated. The blue curves correspond to $p_{0.01,0.02,\ldots,0.1}$, the green curves to $p_{0.2,0.3,\ldots,0.9}$ and the red curves to $p_{0.91,0.92,\ldots,1}$. The C-indicator shown is evaluated for $\alpha_0 = 0.92$ and $\beta = 0.99$. The vertical dotted lines crossing the images indicate a window of acceptable "true" trigger time steps, as identified by a domain expert. For the RCCI cases, the trigger time ranges are based on the High Temperature Heat Release (HTHR), i.e., the second peak in the Heat Release Rate (HRR) profiles.

3.3 Defining a Trigger

In addition to defining a noise-resistant indicator function, we also need to define a trigger function that returns a boolean value, capturing whether a property of the indicator has been met. Looking at Figure 3, we notice that across all experiments from Table 1, the C-indicator is increasing during the true trigger time step windows. Therefore, we seek to find a value $\tau_C \in (0, 1)$, such that $C_{\alpha,\beta}(t)$ crosses τ_C *from below*, as the simulation time t progresses.

Figure 4 plots the trigger time steps as a function of τ_C for $C_{\alpha=0.92,\beta=0.99}(t)$. This plot shows that the viable range of τ_C is $[0.01, 0.05]$ for the four use cases described in Table 1. The horizontal dashed lines indicate the true trigger range identified by our domain expert. We consider those values of τ_C that fall within the horizontal dashed lines to be good τ_C values for our trigger, with those values of τ_C falling below the dashed lines still considered viable. We note the plots for other values of α and β look similar and have been omitted due to lack of space in this text.

4. COMPUTING INDICATORS AND TRIGGERS EFFICIENTLY: A SUBLINEAR APPROACH

The previous section showed that C-indicator and trigger are robust to noise fluctuations and act as a precursor to rapid heat release in combustion simulations. In [3] we show the computational cost of computing CEMA-based indicators can be prohibitive for large-scale simulations (up to 60 times the cost of a simulation time-step depending on the simulation parameters). To mitigate the cost, we introduced a quantile sampling approach that comes with provable bounds on accuracy as a function of the number of samples. Most importantly, the required number of sam-

Table 1: Four Combustion Use Cases analyzed in this study. The "true" trigger time ranges are estimated based on $95-100^{\text{th}}$ percentiles of the heat release rate. The computed time ranges were evaluated using our quantile sampling approach. The C-trigger and P-trigger were computed over 50 realizations of experiments with 20 samples per process.

Problem Instance	Number of Grid Points	Number of Species	"True" Trigger Time Range	C-trigger Detection	P-trigger Detection [3]	Total Processes
HCCI, T=15	451,584	28	250-315	235 - 265	250 - 262	1600
HCCI, T=40	451,584	28	175-225	180 - 220	213 - 220	1600
RCCI, case 6	2,560K	116	38-50	34 - 38	28 - 45	6400
RCCI, case 7	2,560K–10,240K	116	42-58	32.5 - 35.5	35 - 50	6400

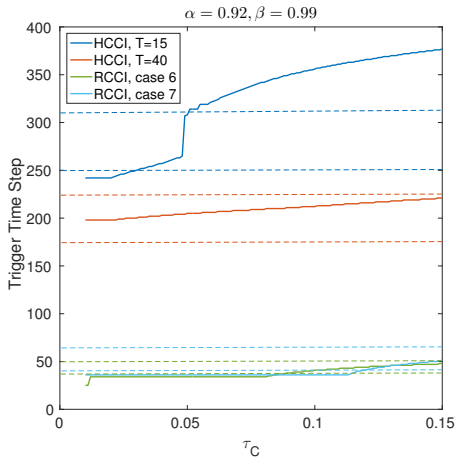

Figure 4: This figure plots the trigger time steps as a function of τ_C for $C_{\alpha=0.92,\beta=0.99}(t)$. There is a range of viable values of $\tau_C \in [0.0, 0.05]$ that predict early stage heat release.

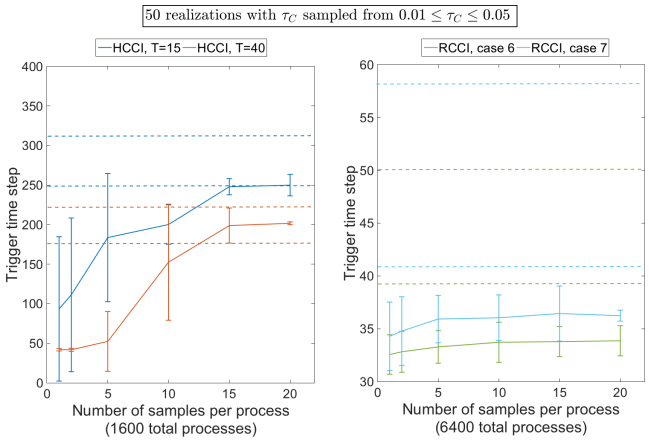

Figure 5: Plots illustrating the variability of the trigger time steps predicted by the C-indicator and trigger as a function of the number of samples per process. The data for these plots was generated via 50 realizations of the C-indicator with $\alpha = 0.92$ and $\beta = 0.99$, and τ_C drawn from $[0.01, 0.05]$.The horizontal dashed lines define the range of time steps within which we would like to make the workflow transition (as identified by a domain expert.

ples for a specified accuracy is independent of the size of the problem, hence our sampling based algorithms offer excellent scalability.

We summarize the quantile sampling method here and refer the reader to [3] for analytical and empirical proofs of convergence. Consider an array $A \in \mathbb{R}^N$, in sorted order. Our aim is to estimate the α-percentile of A. (We use p_α to denote the percentiles.) Note that this is exactly the entry $A_{\lceil \alpha N \rceil}$. Here is a simple sampling procedure.

1. Sample k independent, uniform indices $r_1, r_2, \ldots, r_k$ in $\{1, 2, \ldots, N\}$.
   Denote by $\widehat{A}$ the sorted array $[A(r_1), A(r_2), \ldots, A(r_k)]$.
2. Output the α-percentile of $\widehat{A}$ as the estimate, $\widehat{p}_\alpha$.

We performed a series of experiments examining the variation in the trigger time steps as a function of the number of samples used per process. The data for Figure 5 was generated via 50 realizations of the C-indicator with $\alpha = 0.92$ and $\beta = 0.99$, with τ_C drawn from $[0.01, 0.05]$. The horizontal dashed lines in this figure define the range of true trigger time steps within which we would like to make the workflow transition (as identified by a domain expert). This plot demonstrates that, even across the range of τ_C values, with a small number of samples per process, our quantile sampling approach can accurately estimate the true trigger time steps as defined by the domain expert. The accuracy of our quantile-based sampling predictions is also shown in Table 1, which lists the the triggers predicted by the C-indicator and P-indicator that we used in our previous work [3]. The P-indicator is computed as $P_{\alpha,\beta,\gamma} = \frac{p_\alpha - p_\gamma}{p_\beta - p_\gamma}$, for $\alpha = 0.94$, $\beta = 0.98$, $\gamma = 0.01$, and the trigger threshold chosen as $[0.725, 0.885]$. The results show both methods are accurate. We are currently investigating how we can improve our results by using two metrics concurrently. Note that using both metrics will not induce an additional burden, since the two metrics can be computed from the same set of samples.

5. CONCLUSION

We propose a new indicator and trigger for making data-driven control-flow decisions in situ. Using a provably robust sampling-based approach, of CEMA values (which, when computed in full, can cost up to 60 times a simulation time step). Our experiments show that our proposed indicator, based on the coefficient of variation, can efficiently predict rapid increases in heat release. We believe the new metric can be deployed collectively with a previously defined mechanism, to provide more robust detections, since together they will be less prone to false positives. We note that using both metrics will not induce an additional burden, since the two metrics can be computed from the same set of samples. Our future work aims to explore deployment of both metrics jointly in production simulation runs.

6. REFERENCES

[1] H. Abbasi, G. Eisenhauer, M. Wolf, K. Schwan, and S. Klasky. Just In Time: Adding Value to The IO Pipelines of High Performance Applications with JITStaging. In *Proc. of 20th International Symposium on High Performance Distributed Computing (HPDC'11)*, June 2011.

[2] J. C. Bennett, H. Abbasi, P.-T. Bremer, R. Grout, A. Gyulassy, T. Jin, S. Klasky, H. Kolla, M. Parashar, V. Pascucci, P. Pebay, D. Thompson, H. Yu, F. Zhang, and J. Chen. Combining in-situ and in-transit processing to enable extreme-scale scientific analysis. In J. Hollingsworth, editor, *SC '12: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis, Salt Lake Convention Center, Salt Lake City, UT, USA, November 10–16, 2012*, pages 49:1–49:9, pub-IEEE:adr, 2012. IEEE Computer Society Press.

[3] J. C. Bennett, A. Bhagatwala, J. H. Chen, C. Seshadhri, A. Pinar, and M. Salloum. Trigger detection for adaptive scientific workflows using percentile sampling. Technical report, 2015. arXiv:1506.08258.

[4] A. Bhagatwala, J. H. Chen, and T. Lu. Direct numerical simulations of SACI/HCCI with ethanol. *Comb. Flame*, 161:1826–1841, 2014.

[5] A. Bhagatwala, R. Sankaran, S. Kokjohn, and J. H. Chen. Numerical investigation of spontaneous flame propagation under RCCI conditions. *Comb. Flame*, Under review.

[6] J.-M. F. Brad Whitlock and J. S. Meredith. Parallel In Situ Coupling of Simulation with a Fully Featured Visualization System. In *Proc. of 11th Eurographics Symposium on Parallel Graphics and Visualization (EGPGV'11)*, April 2011.

[7] J. H. Chen, A. Choudhary, B. de Supinski, M. DeVries, E. R. Hawkes, S. Klasky, W. K. Liao, K. L. Ma, J. Mellor-Crummey, N. Podhorski, R. Sankaran, S. Shende, and C. S. Yoo. Terascale direct numerical simulations of turbulent combustion using S3D. *Computational Science and Discovery*, 2:1–31, 2009.

[8] N. Fabian, K. Moreland, D. Thompson, A. Bauer, P. Marion, B. Gevecik, M. Rasquin, and K. Jansen. The paraview coprocessing library: A scalable, general purpose in situ visualization library. In *Proc. of IEEE Symposium on Large Data Analysis and Visualization (LDAV)*, pages 89 –96, October 2011.

[9] S. L. Kokjohn, R. M. Hanson, D. A. Splitter, and R. D. Reitz. Fuel reactivity controlled compression ignition (rcci): A pathway to controlled high-efficiency clean combustion. *Int. J. Engine. Res.*, 12:209–226, 2011.

[10] T. Lu, C. S. Yoo, J. H. Chen, and C. K. Law. Three-dimensional direct numerical simulation of a turbulent lifted hydrogen flame in heated coflow: a chemical explosive mode analysis. *J. Fluid Mech.*, 652:45–64, 2010.

[11] K. Myers, E. Lawrence, M. Fugate, J. Woodring, J. Wendelberger, and J. Ahrens. An in situ approach for approximating complex computer simulations and identifying important time steps. Technical report, 2014. arXiv:1409.0909v1.

[12] B. Nouanesengsy, J. Woodring, J. Patchett, K. Myers, and J. Ahrens. ADR visualization: A generalized framework for ranking large-scale scientific data using analysis-driven refinement. In *Large Data Analysis and Visualization (LDAV), 2014 IEEE 4th Symposium on*, pages 43–50. IEEE, 2014.

[13] R. Shan, C. S. Yoo, J. H. Chen, and T. Lu. Computational diagnostics for n-heptane flames with chemical explosive mode analysis. *Comb. Flame*, 159:3119–3127, 2012.

[14] V. Vishwanath, M. Hereld, and M. Papka. Toward simulation-time data analysis and i/o acceleration on leadership-class systems. In *Proc. of IEEE Symposium on Large Data Analysis and Visualization (LDAV)*, October 2011.

[15] H. Yu, C. Wang, R. W. Grout, J. H. Chen, and K.-L. Ma. In-situ visualization for large-scale combustion simulations. *IEEE Computer Graphics and Applications*, 30:45–57, 2010.